Additional Task

Name:Shaik Khaja Basha
Batch : Batch 11
Date : 24.07.2025
Task : Meta Arguments

Meta Arguments

In Terraform, **Meta-Arguments** are special arguments that control **how resources behave**, rather than defining the resource itself. These are not specific to any provider and can be used with most resources.

Dependency

In Terraform, a dependency defines the order in which resources are created or destroyed. Terraform builds a dependency graph internally based on how resources reference each other. This ensures resources are created/destroyed in the correct order.

Types of Dependencies

1. Implicit/direct Dependencies (Automatic)

- Terraform automatically detects dependencies when you reference attributes from other resources.
- It is use as reference to other resources

Example:

I have resources like

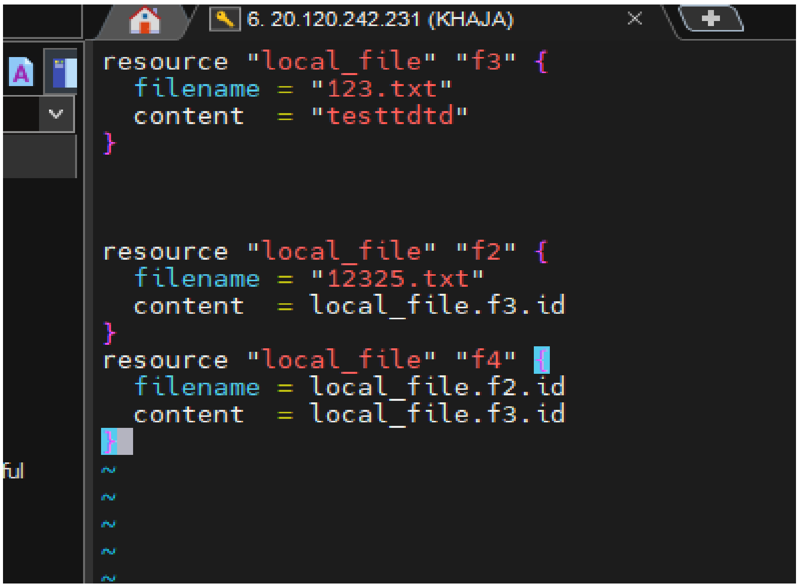
```
resource "local_file" "f3" {
  filename = "123.txt"
  content  = "test"
}
resource "local_file" "f2" {
  filename = "12325.txt"
  content  = local_file.f3.id   # here we are using id of first resource
}
```

```
resource "local_file" "f4" {
  filename = local_file.f2.id
  content  = local_file.f3.id


}


resource "local_file" "f1" {
  filename = "15342541"
  content  = "sgfj"
}
```

- Here we are using direct resource form 1st resource to execute the 2nd resource by using the id of the first resource as shown above
- We will get id for the second resource when the first resource is executed and it’s a chain process which depend on each other

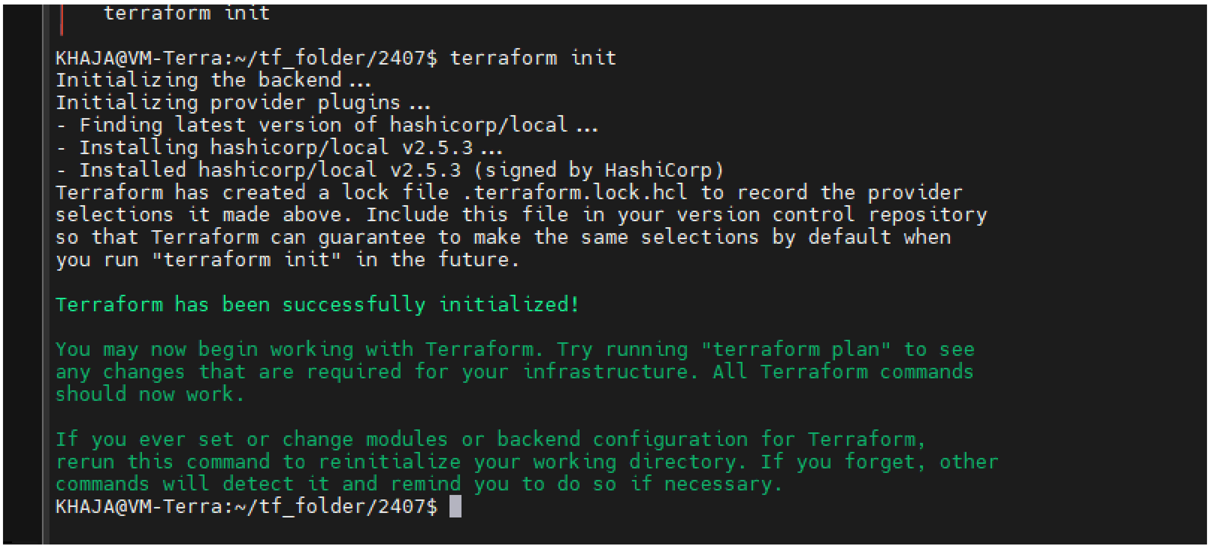
Creating of resources in res.tf file

```
resource "local_file" "f3" {
  filename = "123.txt"
  content  = "testtdtd"
}
resource "local_file" "f2" {
  filename = "12325.txt"
  content  = local_file.f3.id
}
resource "local_file" "f4" {
  filename = local_file.f2.id
  content  = local_file.f3.id
}
```



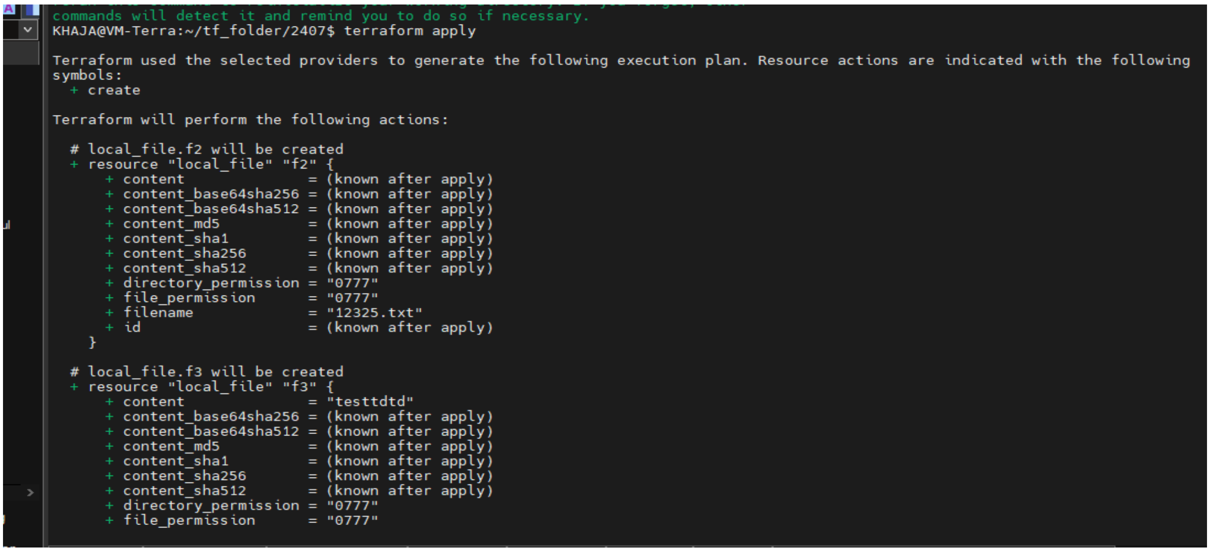
```
resource "local_file" "f3" {
  filename = "123.txt"
  content  = "testtdtd"
}



resource "local_file" "f2" {
  filename = "12325.txt"
  content  = local_file.f3.id
}
resource "local_file" "f4" {
  filename = local_file.f2.id
  content  = local_file.f3.id
}
~
~
~
~
~
```

Execute the command terraform init

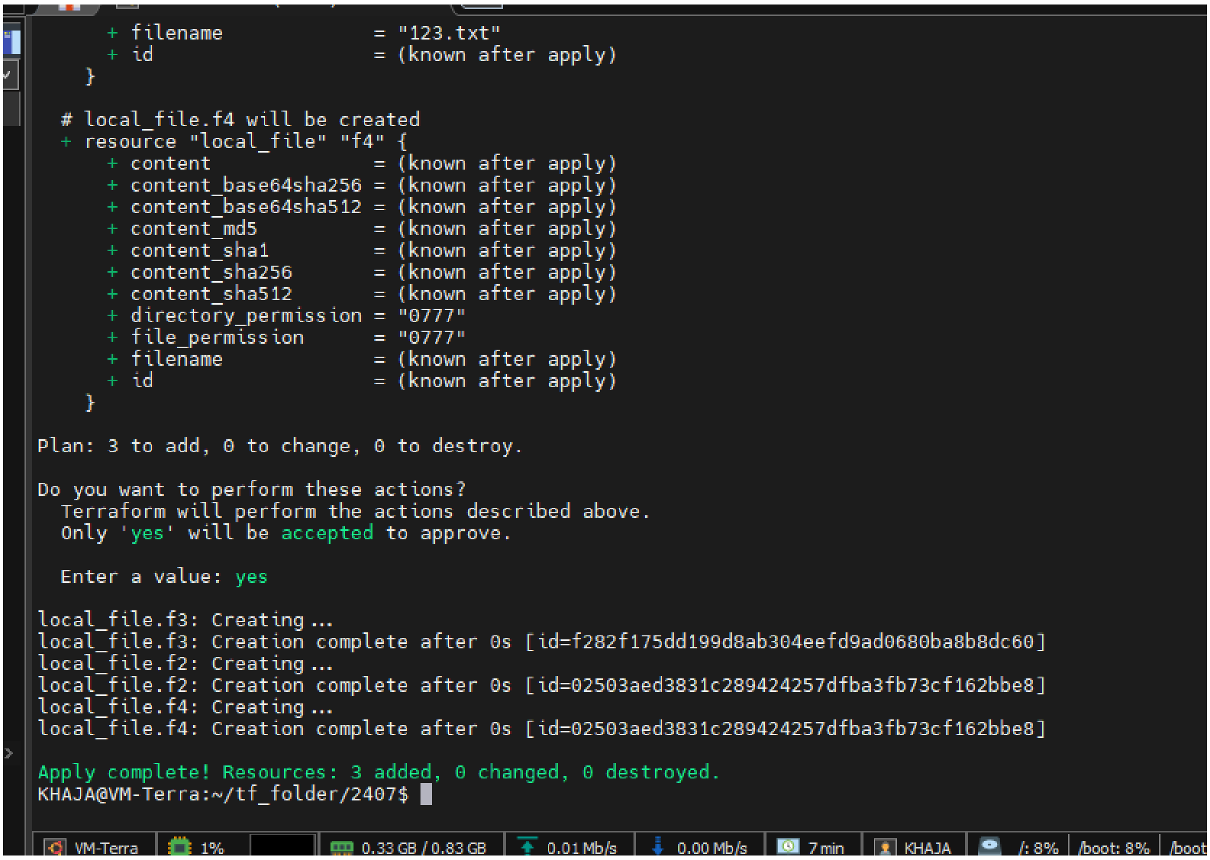


```
    terraform init

KHAJA@VM-Terra:~/tf_folder/2407$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
KHAJA@VM-Terra:~/tf_folder/2407$ 
```

Execute the terraform apply to make changes



```
commands will detect it and remind you to do so if necessary.
KHAJA@VM-Terra:~/tf_folder/2407$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # local_file.f2 will be created
  + resource "local_file" "f2" {
      + content              = (known after apply)
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "12325.txt"
      + id                   = (known after apply)
    }

  # local_file.f3 will be created
  + resource "local_file" "f3" {
      + content              = "testtdtd"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
```

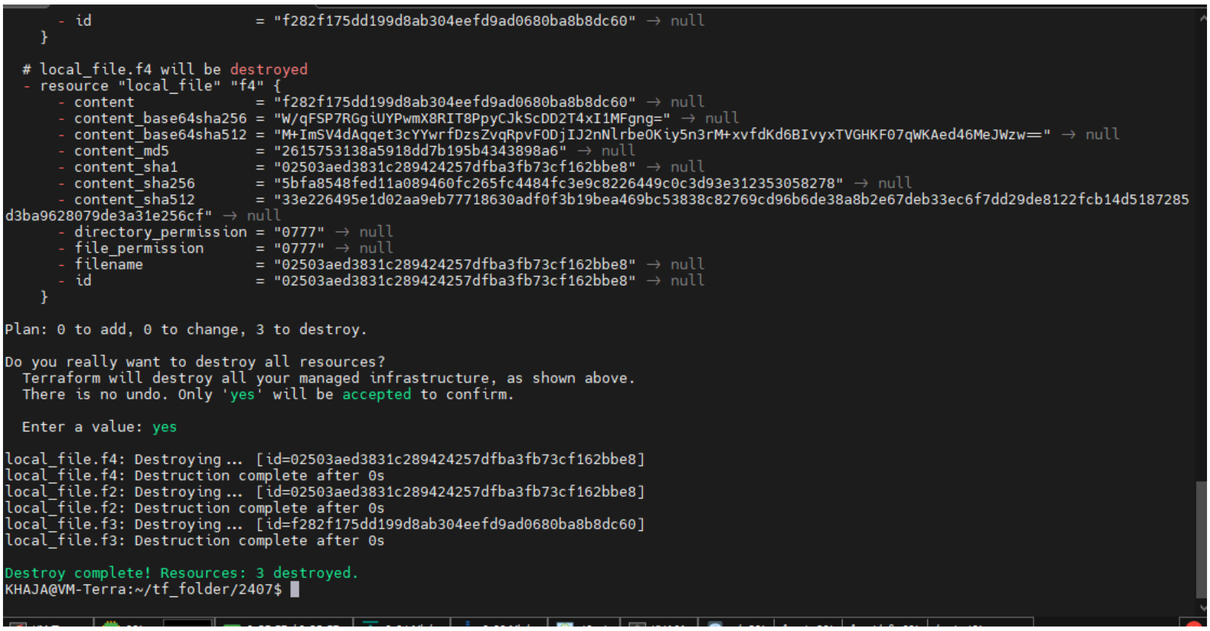


```
      + filename             = "123.txt"
      + id                   = (known after apply)
    }

  # local_file.f4 will be created
  + resource "local_file" "f4" {
      + content              = (known after apply)
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = (known after apply)
      + id                   = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.f3: Creating...
local_file.f3: Creation complete after 0s [id=f282f175dd199d8ab304eefd9ad0680ba8b8dc60]
local_file.f2: Creating...
local_file.f2: Creation complete after 0s [id=02503aed3831c289424257dfba3fb73cf162bbe8]
local_file.f4: Creating...
local_file.f4: Creation complete after 0s [id=02503aed3831c289424257dfba3fb73cf162bbe8]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$ 
```

Form the output we can see after creating of file.f3 its executing file.f2 which depend on f3

And like that for f4

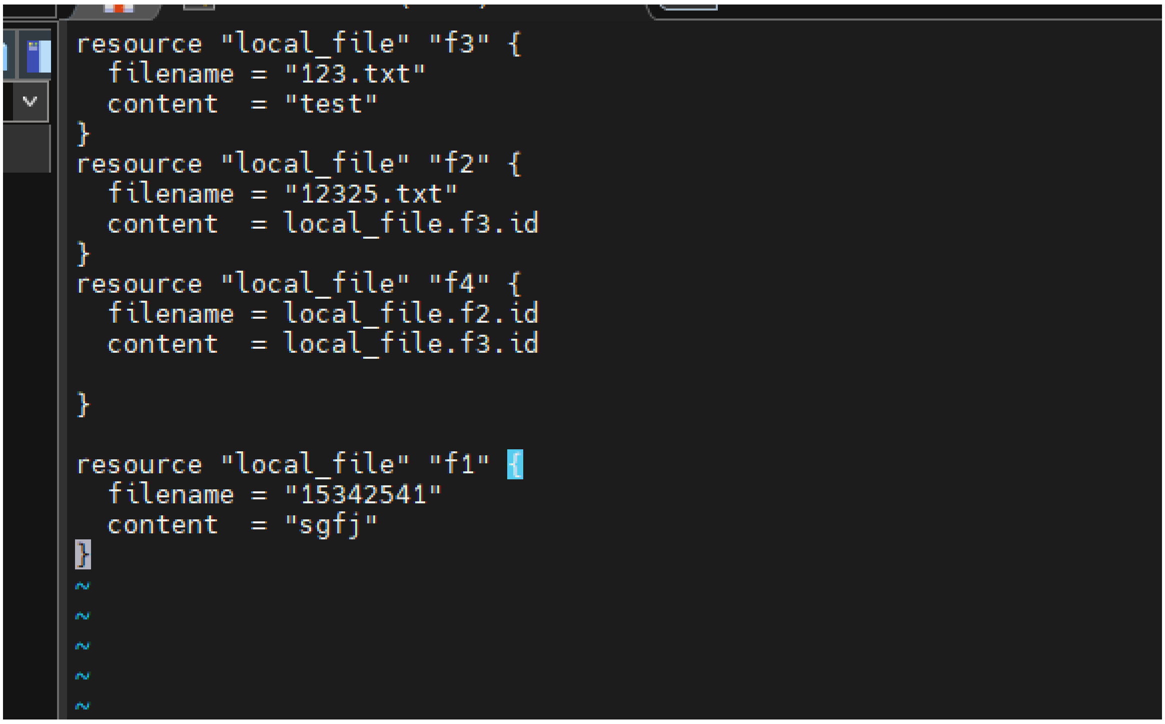
Lets destroy it



```
      - id                   = "f282f175dd199d8ab304eefd9ad0680ba8b8dc60" -> null
    }

  # local_file.f4 will be destroyed
  - resource "local_file" "f4" {
      - content              = "f282f175dd199d8ab304eefd9ad0680ba8b8dc60" -> null
      - content_base64sha256 = "W/qFSP7RGgiUYPwmX8RIT8PpyCJkScDD2T4xI1MFgng=" -> null
      - content_base64sha512 = "M+ImSV4dAqqet3cYYwrfDzsZvqRpvFODjIJ2nNlrbeOKiy5n3rM+xvfdKd6BIvyxTVGHKF07qWKAed46MeJWzw==" -> null
      - content_md5          = "2615753138a5918dd7b195b4343898a6" -> null
      - content_sha1         = "02503aed3831c289424257dfba3fb73cf162bbe8" -> null
      - content_sha256       = "5bfa8548fed11a089460fc265fc4484fc3e9c8226449c0c3d93e312353058278" -> null
      - content_sha512       = "33e226495e1d02aa9eb77718630adf0f3b19bea469bc53838c82769cd96b6de38a8b2e67deb33ec6f7dd29de8122fcb14d5187285
d3ba9628079de3a31e256cf" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "02503aed3831c289424257dfba3fb73cf162bbe8" -> null
      - id                   = "02503aed3831c289424257dfba3fb73cf162bbe8" -> null
    }

Plan: 0 to add, 0 to change, 3 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

local_file.f4: Destroying... [id=02503aed3831c289424257dfba3fb73cf162bbe8]
local_file.f4: Destruction complete after 0s
local_file.f2: Destroying... [id=02503aed3831c289424257dfba3fb73cf162bbe8]
local_file.f2: Destruction complete after 0s
local_file.f3: Destroying... [id=f282f175dd199d8ab304eefd9ad0680ba8b8dc60]
local_file.f3: Destruction complete after 0s

Destroy complete! Resources: 3 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$ 
```

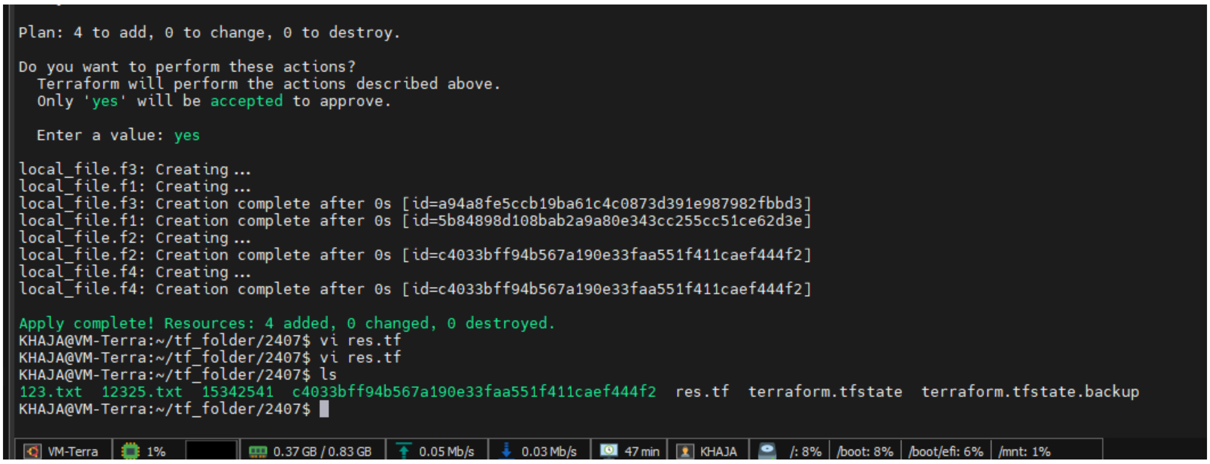
Which destroying also it do the same first its deleting the file f4 then the remaining

Dependency configuration



```
resource "local_file" "f3" {
  filename = "123.txt"
  content  = "test"
}
resource "local_file" "f2" {
  filename = "12325.txt"
  content  = local_file.f3.id
}
resource "local_file" "f4" {
  filename = local_file.f2.id
  content  = local_file.f3.id

}

resource "local_file" "f1" {
  filename = "15342541"
  content  = "sgfj"
}
~
~
~
~
~
```

Here f1 and f3 are independent and other are dependency resource



```
Plan: 4 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.f3: Creating...
local_file.f1: Creating...
local_file.f3: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f1: Creation complete after 0s [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f2: Creating...
local_file.f2: Creation complete after 0s [id=c4033bff94b567a190e33faa551f411caef444f2]
local_file.f4: Creating...
local_file.f4: Creation complete after 0s [id=c4033bff94b567a190e33faa551f411caef444f2]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$ vi res.tf
KHAJA@VM-Terra:~/tf_folder/2407$ vi res.tf
KHAJA@VM-Terra:~/tf_folder/2407$ ls
123.txt  12325.txt  15342541  c4033bff94b567a190e33faa551f411caef444f2  res.tf  terraform.tfstate  terraform.tfstate.backup
KHAJA@VM-Terra:~/tf_folder/2407$ 
```

2. Explicit/indirect Dependency *(Using depends_on)*

Use depends_on when Terraform **can't detect the dependency automatically**, such as when resources are connected only via provisioners or indirect outputs.

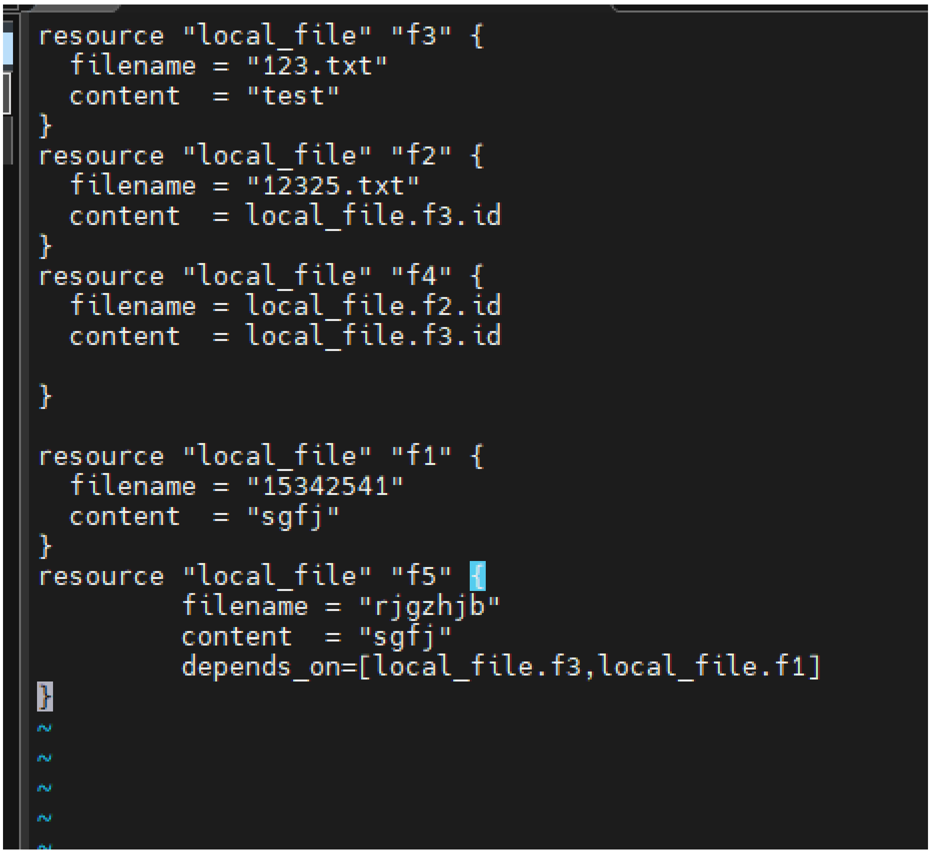
- we will be mentioning the resources as dependencies

example:

lets take resources like

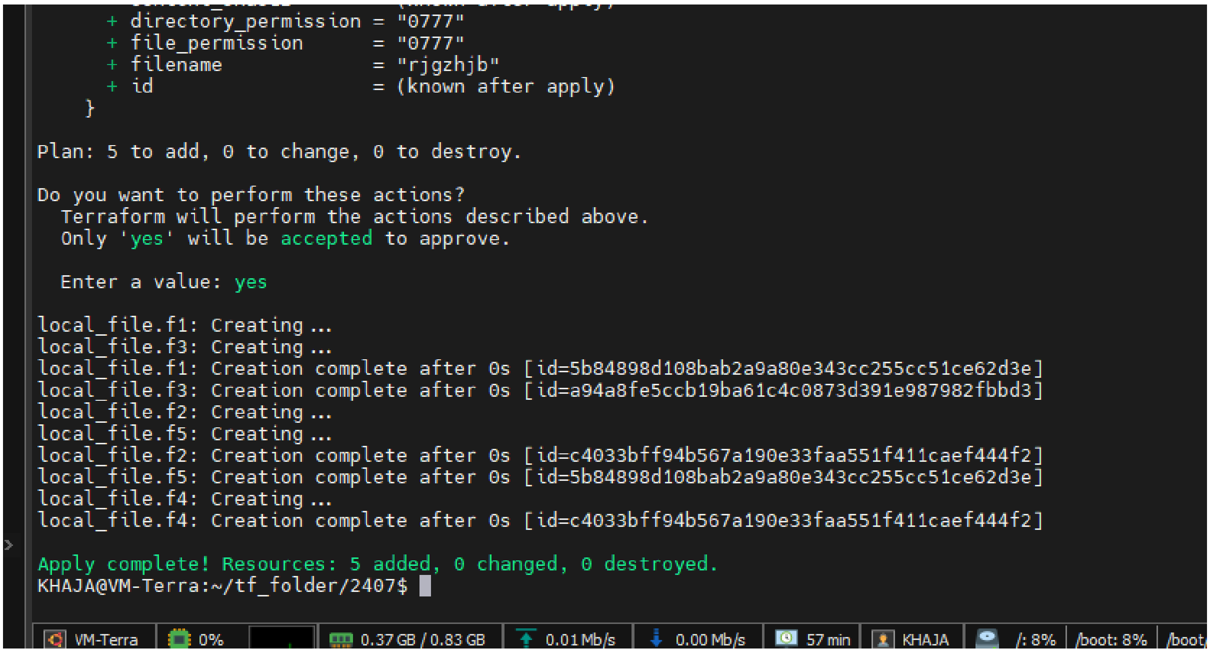
```
resource "local_file" "f3" {

            filename = "123.txt"

            content  = "test"

                     }


resource "local_file" "f1" {

            filename = "15342541"

            content  = "sgfj"

          }

          resource "local_file" "f5" {

          filename = "rjgzhjb"

           content  = "sgfj"

            depends_on=[local_file.f3,local_file.f1]

          }
```

Here f5 will execute when f3 and f1 are executed until it will not executed



```
resource "local_file" "f3" {
  filename = "123.txt"
  content  = "test"
}
resource "local_file" "f2" {
  filename = "12325.txt"
  content  = local_file.f3.id
}
resource "local_file" "f4" {
  filename = local_file.f2.id
  content  = local_file.f3.id

}

resource "local_file" "f1" {
  filename = "15342541"
  content  = "sgfj"
}
resource "local_file" "f5" {
         filename = "rjgzhjb"
         content  = "sgfj"
         depends_on=[local_file.f3,local_file.f1]
}
~
~
~
~
```

Terraform apply



```
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "rjgzhjb"
      + id                   = (known after apply)
    }

Plan: 5 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.f1: Creating...
local_file.f3: Creating...
local_file.f1: Creation complete after 0s [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f3: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f2: Creating...
local_file.f5: Creating...
local_file.f2: Creation complete after 0s [id=c4033bff94b567a190e33faa551f411caef444f2]
local_file.f5: Creation complete after 0s [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f4: Creating...
local_file.f4: Creation complete after 0s [id=c4033bff94b567a190e33faa551f411caef444f2]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$ 
```

Here we can see once f1 and f3 are created then f5 is created bcz its depend on both the files

Lifecycle:

- Used when changing one resource should force replacement of another, even if it’s not directly referenced.
- destroy and then create

create before destroy: Zero-downtime replacements:

use case

When enabled, Terraform first creates the new resource before destroying the old one — which helps avoid downtime for services like:

- VMs
- Load balancers
- App gateways

Useful when a field change would normally force resource replacement.

Syntax:

```
lifecycle {
    create_before_destroy = true
    }
```

Prevent_destroy: A lifecycle rule that blocks resource deletion in Terraform, even during terraform destroy or resource replacement.

- creates a lock on the resource and doesn’t allow for modification

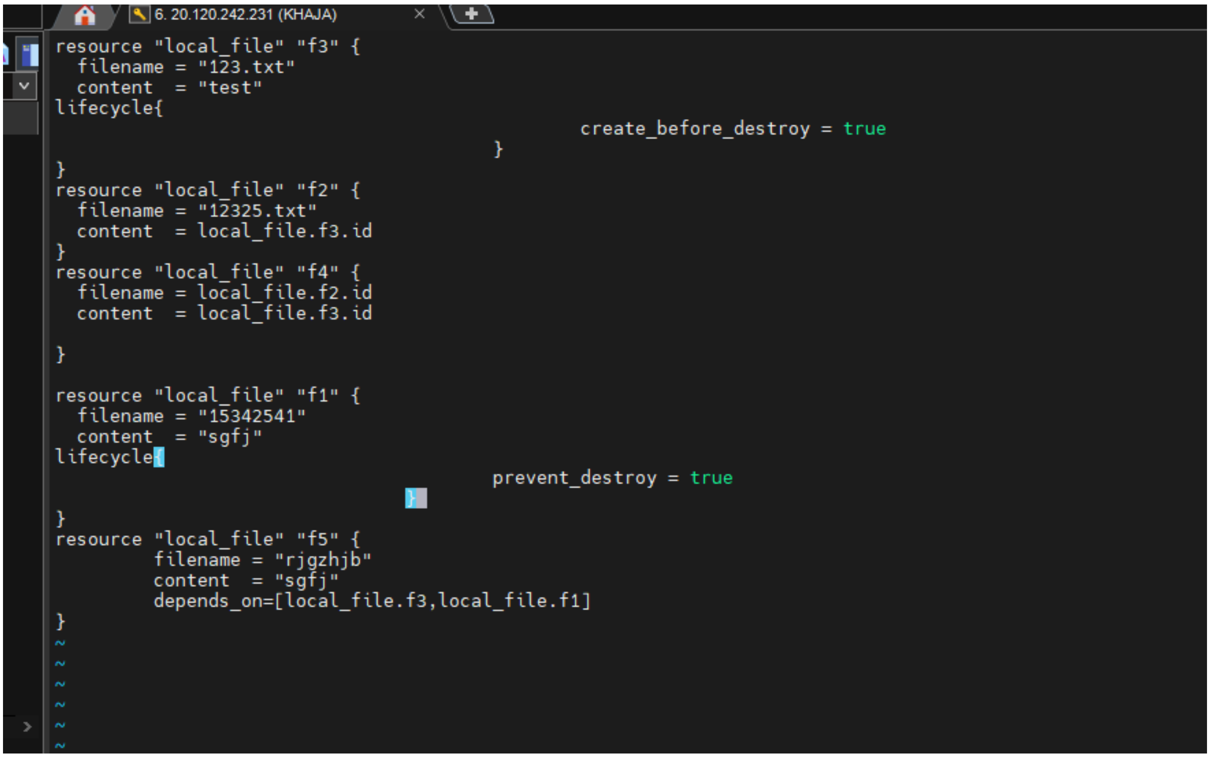
If a resource has prevent_destroy = true:

- Running terraform destroy will **fail** if that resource is in the config
- Useful for critical resources like databases or production VMs

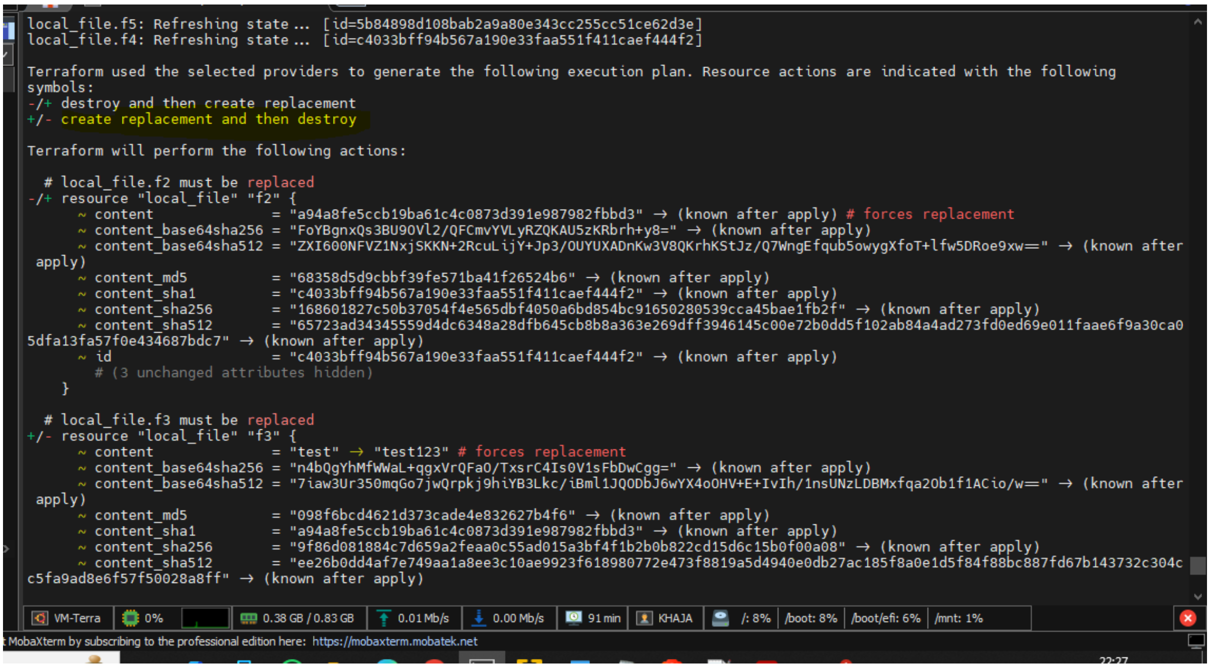
Syntax:

```
    lifecycle {
        prevent_destroy = true
            }
```

Lets use in present resources

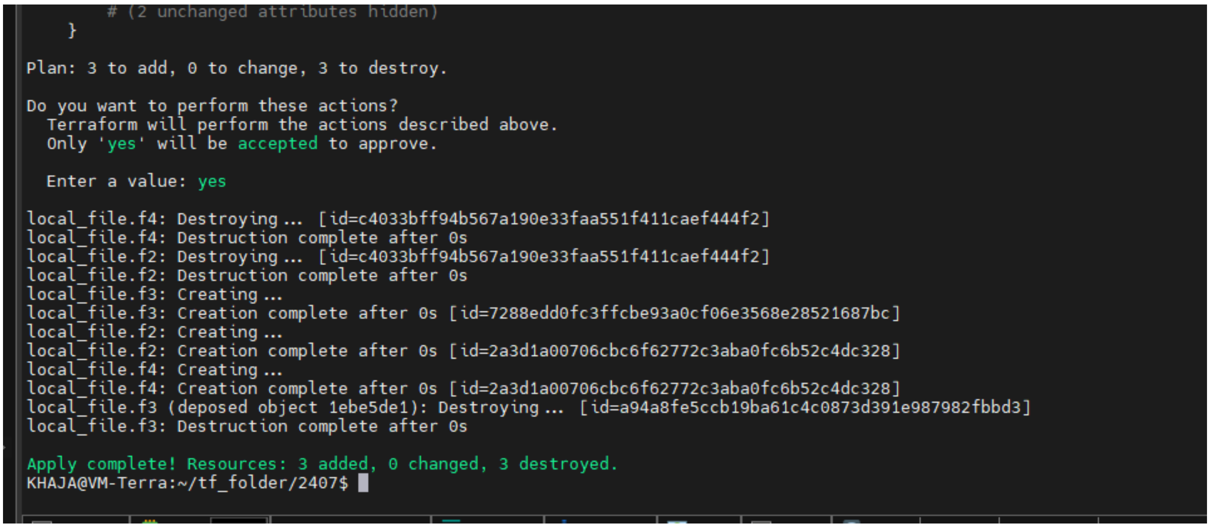


```
resource "local_file" "f3" {
  filename = "123.txt"
  content  = "test"
lifecycle{
                                           create_before_destroy = true
                                   }
}
resource "local_file" "f2" {
  filename = "12325.txt"
  content  = local_file.f3.id
}
resource "local_file" "f4" {
  filename = local_file.f2.id
  content  = local_file.f3.id

}

resource "local_file" "f1" {
  filename = "15342541"
  content  = "sgfj"
lifecycle{
                                   prevent_destroy = true
                           }
}
resource "local_file" "f5" {
         filename = "rjgzhjb"
         content  = "sgfj"
         depends_on=[local_file.f3,local_file.f1]
}
~
~
~
~
~
~
```



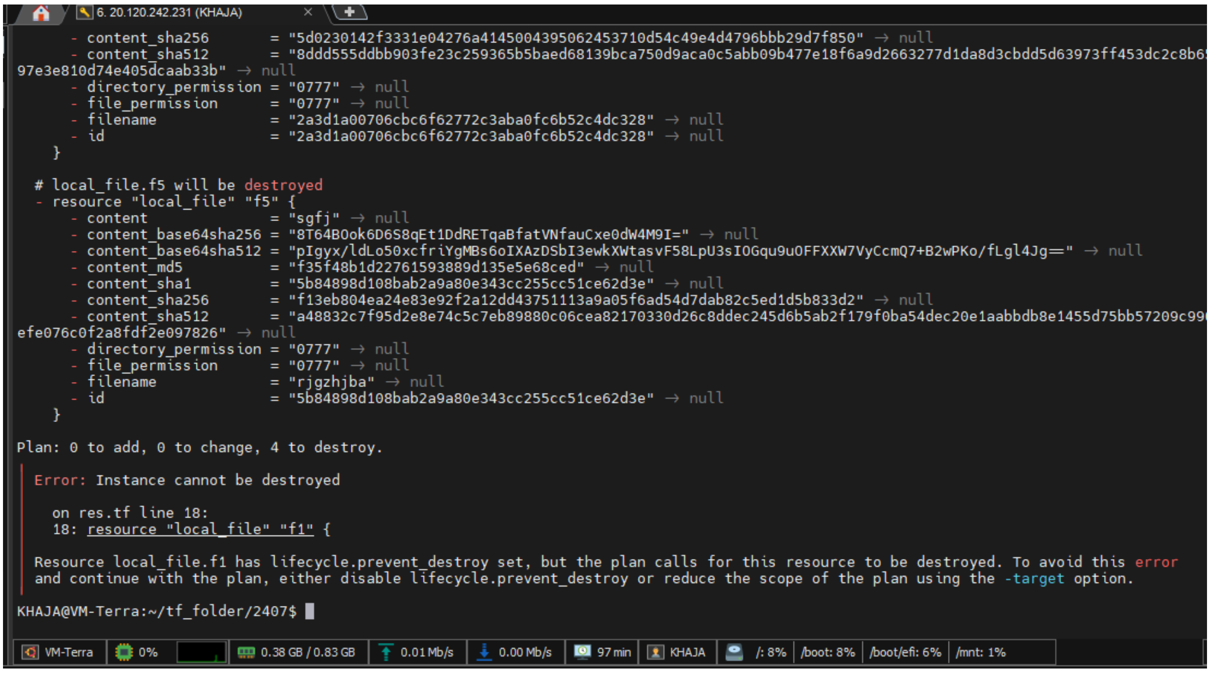
```
local_file.f5: Refreshing state... [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f4: Refreshing state... [id=c4033bff94b567a190e33faa551f411caef444f2]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
-/+ destroy and then create replacement
+/- create replacement and then destroy

Terraform will perform the following actions:

  # local_file.f2 must be replaced
-/+ resource "local_file" "f2" {
      ~ content              = "a94a8fe5ccb19ba61c4c0873d391e987982fbbd3" → (known after apply) # forces replacement
      ~ content_base64sha256 = "FoYBgnxQs3BU90Vl2/QFCmvYVLyRZQKAU5zKRbrh+y8=" → (known after apply)
      ~ content_base64sha512 = "ZXI600NFVZ1NxjSKKN+2RcuLijY+Jp3/0UYUXADnKw3V8QKrhKStJz/Q7WngEfqub5owygXfoT+lfw5DRoe9xw==" → (known after
 apply)
      ~ content_md5          = "68358d5d9cbbf39fe571ba41f26524b6" → (known after apply)
      ~ content_sha1         = "c4033bff94b567a190e33faa551f411caef444f2" → (known after apply)
      ~ content_sha256       = "168601827c50b37054f4e565dbf4050a6bd854bc91650280539cca45bae1fb2f" → (known after apply)
      ~ content_sha512       = "65723ad34345559d4dc6348a28dfb645cb8b8a363e269dff3946145c00e72b0dd5f102ab84a4ad273fd0ed69e011faae6f9a30ca0
5dfa13fa57f0e434687bdc7" → (known after apply)
      ~ id                   = "c4033bff94b567a190e33faa551f411caef444f2" → (known after apply)
        # (3 unchanged attributes hidden)
    }

  # local_file.f3 must be replaced
+/- resource "local_file" "f3" {
      ~ content              = "test" → "test123" # forces replacement
      ~ content_base64sha256 = "n4bQgYhMfWWaL+qgxVrQFa0/TxsrC4Is0V1sFbDwCgg=" → (known after apply)
      ~ content_base64sha512 = "7iaw3Ur350mqGo7jwQrpkj9hiYB3Lkc/iBml1JQODbJ6wYX4oOHV+E+IvIh/1nsUNzLDBMxfqa20b1f1ACio/w==" → (known after
 apply)
      ~ content_md5          = "098f6bcd4621d373cade4e832627b4f6" → (known after apply)
      ~ content_sha1         = "a94a8fe5ccb19ba61c4c0873d391e987982fbbd3" → (known after apply)
      ~ content_sha256       = "9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08" → (known after apply)
      ~ content_sha512       = "ee26b0dd4af7e749aa1a8ee3c10ae9923f618980772e473f8819a5d4940e0db27ac185f8a0e1d5f84f88bc887fd67b143732c304c
c5fa9ad8e6f57f50028a8ff" → (known after apply)
```

Here its showing create replacement before destroying



```
        # (2 unchanged attributes hidden)
    }

Plan: 3 to add, 0 to change, 3 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.f4: Destroying... [id=c4033bff94b567a190e33faa551f411caef444f2]
local_file.f4: Destruction complete after 0s
local_file.f2: Destroying... [id=c4033bff94b567a190e33faa551f411caef444f2]
local_file.f2: Destruction complete after 0s
local_file.f3: Creating...
local_file.f3: Creation complete after 0s [id=7288edd0fc3ffcbe93a0cf06e3568e28521687bc]
local_file.f2: Creating...
local_file.f2: Creation complete after 0s [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]
local_file.f4: Creating...
local_file.f4: Creation complete after 0s [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]
local_file.f3 (deposed object 1ebe5de1): Destroying... [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f3: Destruction complete after 0s

Apply complete! Resources: 3 added, 0 changed, 3 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$
```

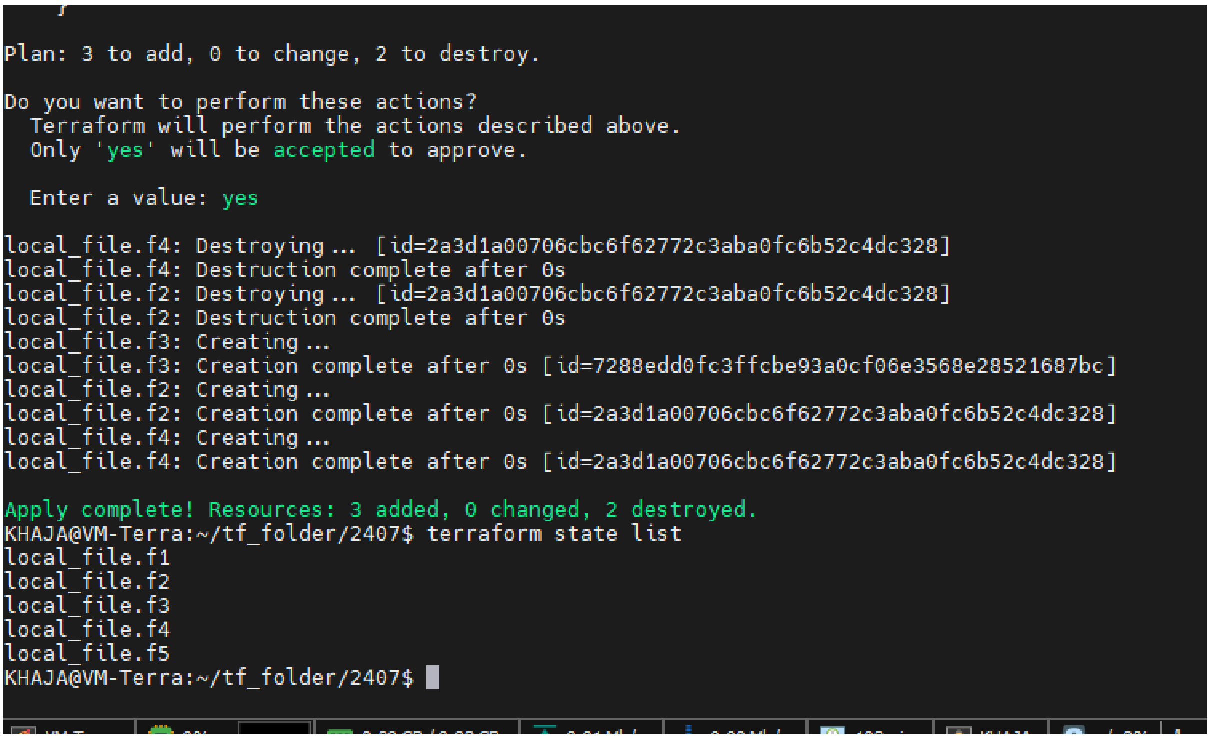
Execute the command terraform destroy



```
      - content_sha256       = "5d0230142f3331e04276a4145004395062453710d54c49e4d4796bbb29d7f850" → null
      - content_sha512       = "8ddd555ddbb903fe23c259365b5baed68139bca750d9aca0c5abb09b477e18f6a9d2663277d1da8d3cbdd5d63973ff453dc2c8b6
97e3e810d74e405dcaab33b" → null
      - directory_permission = "0777" → null
      - file_permission      = "0777" → null
      - filename             = "2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328" → null
      - id                   = "2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328" → null
    }

  # local_file.f5 will be destroyed
  - resource "local_file" "f5" {
      - content              = "sgfj" → null
      - content_base64sha256 = "8T64BOok6D6S8qEt1DdRETqaBfatVNfauCxe0dW4M9I=" → null
      - content_base64sha512 = "pIgyx/ldLo50xcfriYgMBs6oIXAzDSbI3ewkXWtasvF58LpU3sIOGqu9uOFFXXW7VyCcmQ7+B2wPKo/fLgl4Jg==" → null
      - content_md5          = "f35f48b1d22761593889d135e5e68ced" → null
      - content_sha1         = "5b84898d108bab2a9a80e343cc255cc51ce62d3e" → null
      - content_sha256       = "f13eb804ea24e83e92f2a12dd43751113a9a05f6ad54d7dab82c5ed1d5b833d2" → null
      - content_sha512       = "a48832c7f95d2e8e74c5c7eb89880c06cea82170330d26c8ddec245d6b5ab2f179f0ba54dec20e1aabbdb8e1455d75bb57209c99
efe076c0f2a8fdf2e097826" → null
      - directory_permission = "0777" → null
      - file_permission      = "0777" → null
      - filename             = "rjgzhjba" → null
      - id                   = "5b84898d108bab2a9a80e343cc255cc51ce62d3e" → null
    }

Plan: 0 to add, 0 to change, 4 to destroy.

│ Error: Instance cannot be destroyed
│
│   on res.tf line 18:
│   18: resource "local_file" "f1" {
│
│ Resource local_file.f1 has lifecycle.prevent_destroy set, but the plan calls for this resource to be destroyed. To avoid this error
│ and continue with the plan, either disable lifecycle.prevent_destroy or reduce the scope of the plan using the -target option.

KHAJA@VM-Terra:~/tf_folder/2407$ █
```

Its through an error because its have a Lifecyle prevent_before_destroy so it will be not deleted , until we delete manually

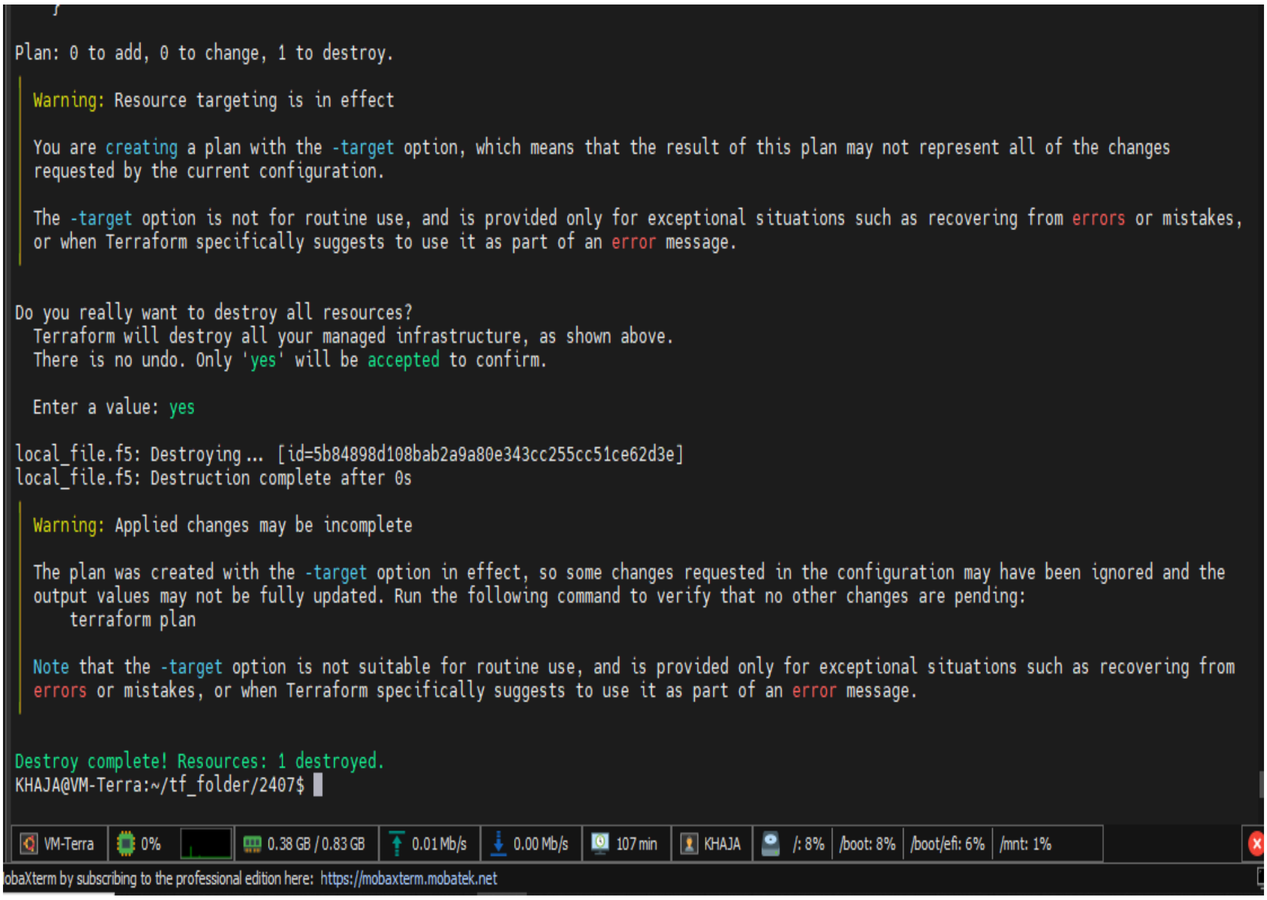
- Even it wont allow us to modify the content



```
Plan: 3 to add, 0 to change, 2 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.f4: Destroying... [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]
local_file.f4: Destruction complete after 0s
local_file.f2: Destroying... [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]
local_file.f2: Destruction complete after 0s
local_file.f3: Creating...
local_file.f3: Creation complete after 0s [id=7288edd0fc3ffcbe93a0cf06e3568e28521687bc]
local_file.f2: Creating...
local_file.f2: Creation complete after 0s [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]
local_file.f4: Creating...
local_file.f4: Creation complete after 0s [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]

Apply complete! Resources: 3 added, 0 changed, 2 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$ terraform state list
local_file.f1
local_file.f2
local_file.f3
local_file.f4
local_file.f5
KHAJA@VM-Terra:~/tf_folder/2407$ █
```

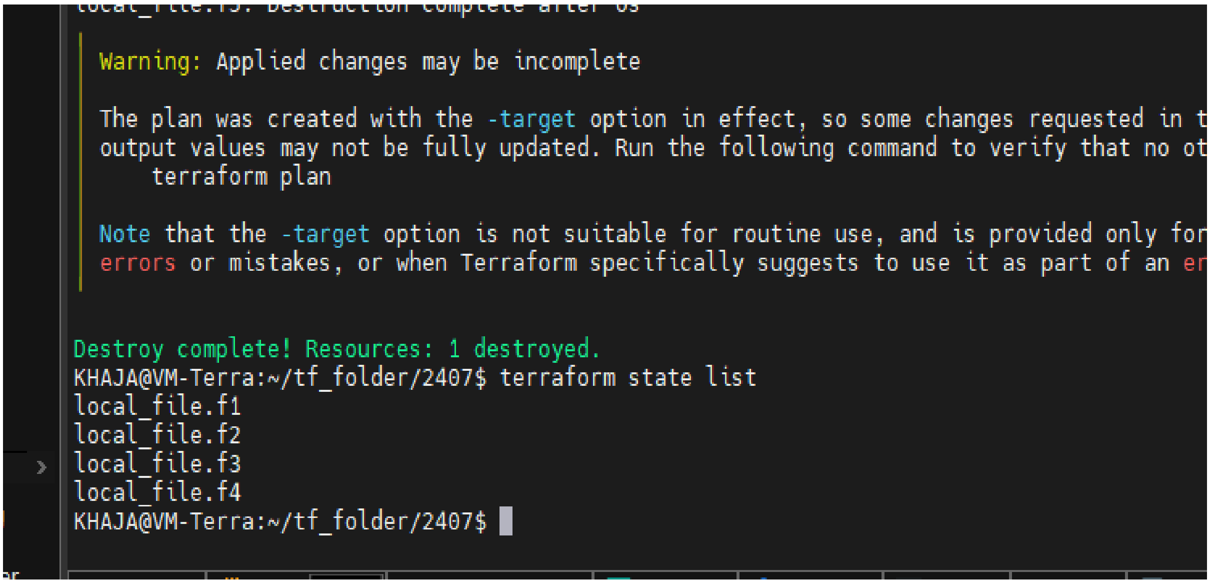
Here I had multiple resource files in state file

Suppose I want to delete specific file I can deleted using **target**



```
Plan: 0 to add, 0 to change, 1 to destroy.

  Warning: Resource targeting is in effect

  You are creating a plan with the -target option, which means that the result of this plan may not represent all of the changes
  requested by the current configuration.

  The -target option is not for routine use, and is provided only for exceptional situations such as recovering from errors or mistakes,
  or when Terraform specifically suggests to use it as part of an error message.


Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

local_file.f5: Destroying... [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f5: Destruction complete after 0s

  Warning: Applied changes may be incomplete

  The plan was created with the -target option in effect, so some changes requested in the configuration may have been ignored and the
  output values may not be fully updated. Run the following command to verify that no other changes are pending:
      terraform plan

  Note that the -target option is not suitable for routine use, and is provided only for exceptional situations such as recovering from
  errors or mistakes, or when Terraform specifically suggests to use it as part of an error message.


Destroy complete! Resources: 1 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$
```

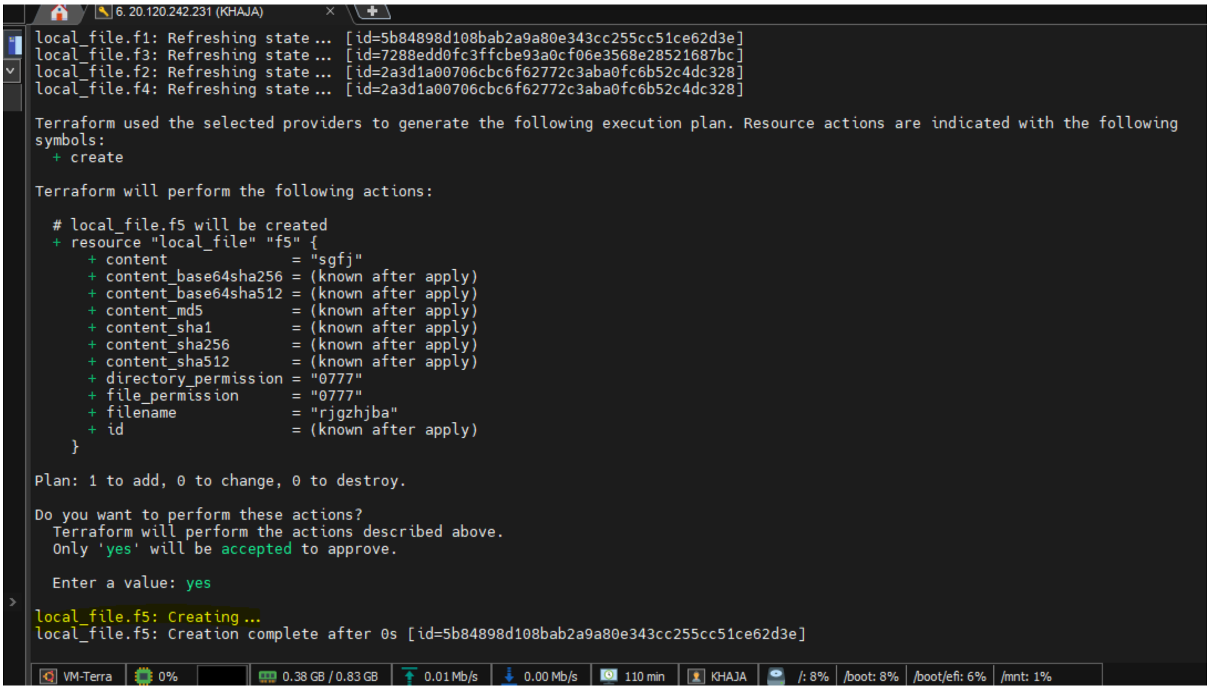
Here we can see explicitly we destroy the define targe file



```
local_file.f5: Destruction complete after 0s

  Warning: Applied changes may be incomplete

  The plan was created with the -target option in effect, so some changes requested in t
  output values may not be fully updated. Run the following command to verify that no ot
      terraform plan

  Note that the -target option is not suitable for routine use, and is provided only for
  errors or mistakes, or when Terraform specifically suggests to use it as part of an er


Destroy complete! Resources: 1 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$ terraform state list
local_file.f1
local_file.f2
local_file.f3
local_file.f4
KHAJA@VM-Terra:~/tf_folder/2407$
```

When we do terraform apply

It will get effected the target delete file



```
local_file.f1: Refreshing state ... [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f3: Refreshing state ... [id=7288edd0fc3ffcbe93a0cf06e3568e28521687bc]
local_file.f2: Refreshing state ... [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]
local_file.f4: Refreshing state ... [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # local_file.f5 will be created
  + resource "local_file" "f5" {
      + content              = "sgfj"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "rjgzhjba"
      + id                   = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.f5: Creating ...
local_file.f5: Creation complete after 0s [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
```

Looping:

1. **count** - Basic Index-Based Looping

Creates multiple resource instances using a counter.

resource "local_file" "f3" {

count = length(var.filename)

filename = var.filename[count.index]

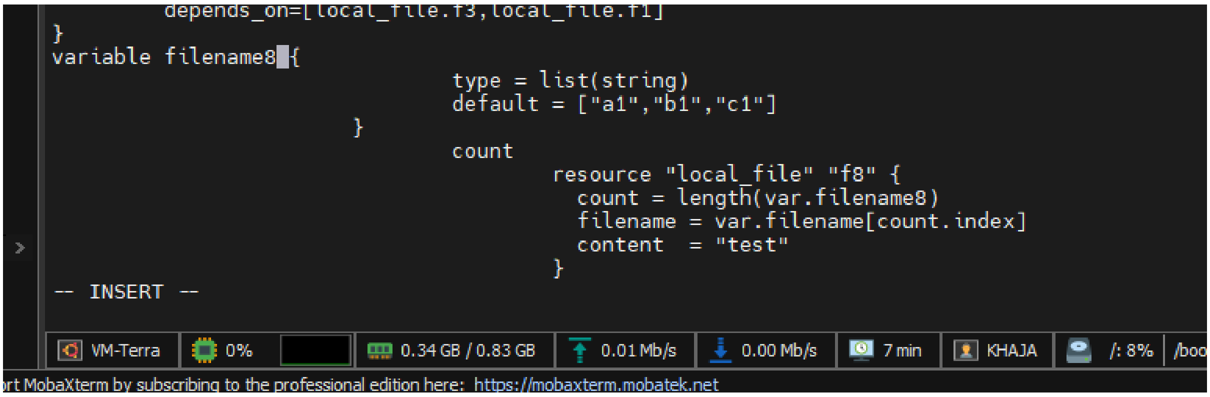
content = "test"

}

When to Use:

- Simple numeric iteration
- Identical resources with numeric suffixes

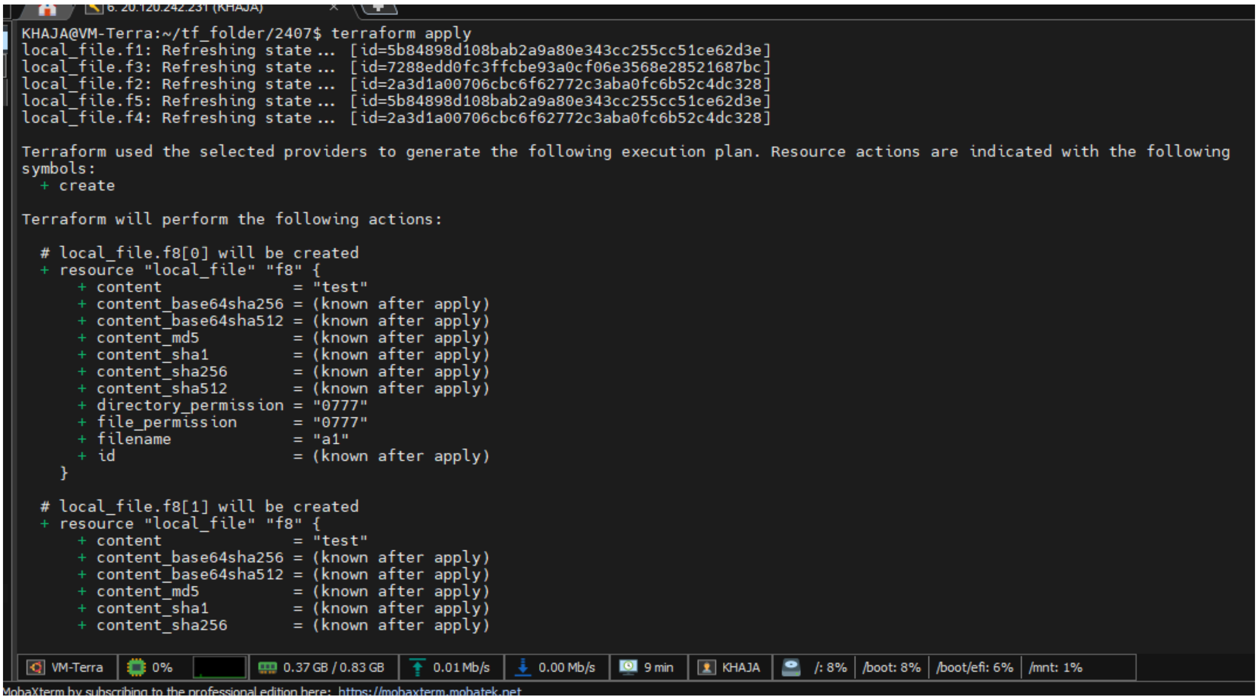
Let create a resource and variable practically



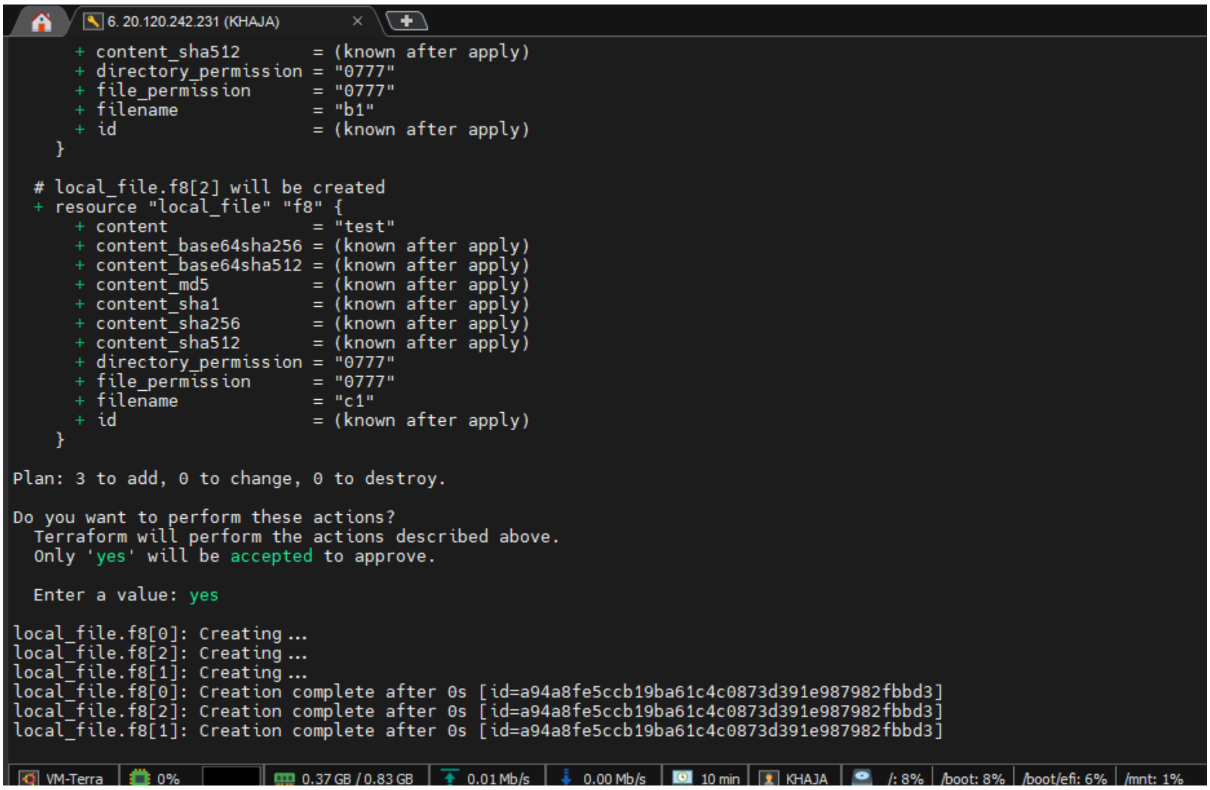
```
         depends_on=[local_file.f3,local_file.f1]
}
variable filename8 {
                                type = list(string)
                                default = ["a1","b1","c1"]
                        }
                                count
                                        resource "local_file" "f8" {
                                          count = length(var.filename8)
                                          filename = var.filename[count.index]
                                          content  = "test"
                                        }

-- INSERT --
```

VM-Terra
0%
0.34 GB / 0.83 GB
0.01 Mb/s
0.00 Mb/s
7 min
KHAJA
/: 8%
/boo
rt MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

Execute the command terraform apply

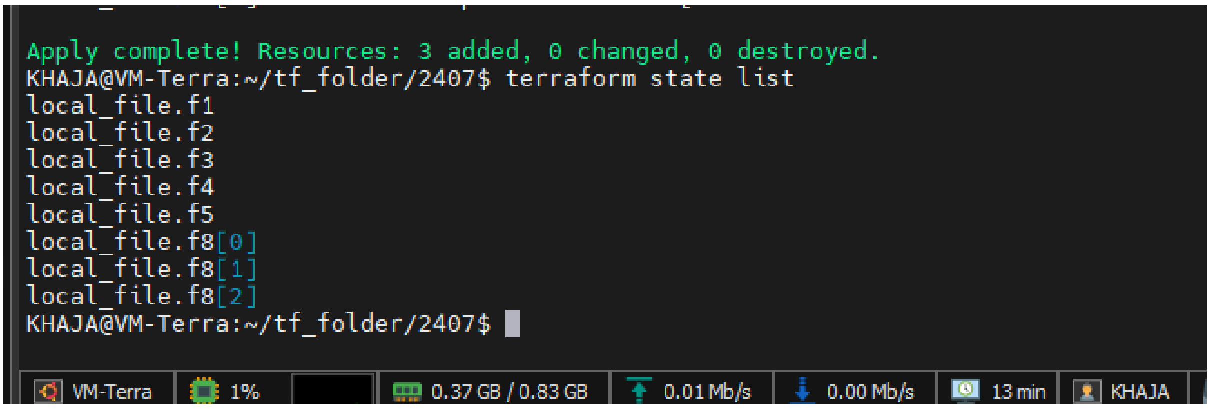


```
KHAJA@VM-Terra:~/tf_folder/2407$ terraform apply
local_file.f1: Refreshing state... [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f3: Refreshing state... [id=7288edd0fc3ffcbe93a0cf06e3568e28521687bc]
local_file.f2: Refreshing state... [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]
local_file.f5: Refreshing state... [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f4: Refreshing state... [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # local_file.f8[0] will be created
  + resource "local_file" "f8" {
      + content              = "test"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "a1"
      + id                   = (known after apply)
    }

  # local_file.f8[1] will be created
  + resource "local_file" "f8" {
      + content              = "test"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
```



```
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "b1"
      + id                   = (known after apply)
    }

  # local_file.f8[2] will be created
  + resource "local_file" "f8" {
      + content              = "test"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "c1"
      + id                   = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.f8[0]: Creating...
local_file.f8[2]: Creating...
local_file.f8[1]: Creating...
local_file.f8[0]: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f8[2]: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f8[1]: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
```

It will be creating along with index



```
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$ terraform state list
local_file.f1
local_file.f2
local_file.f3
local_file.f4
local_file.f5
local_file.f8[0]
local_file.f8[1]
local_file.f8[2]
KHAJA@VM-Terra:~/tf_folder/2407$
```

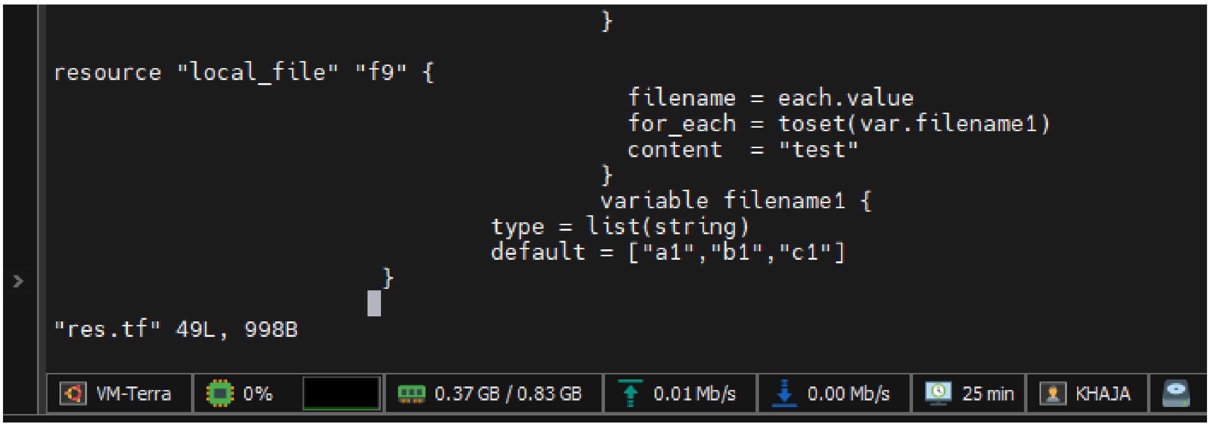
2. **for_each -** Map/Set-Based Looping

Creates resources from a map or set, preserving unique identifiers.

- Its an element base

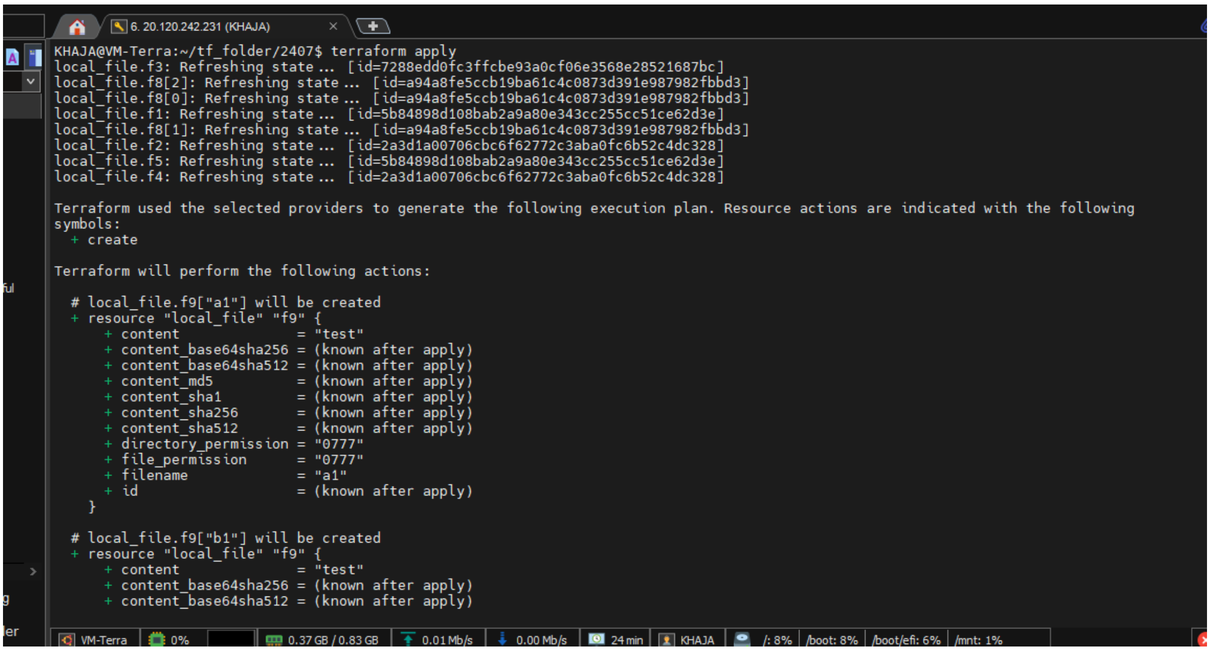
for_each

```
resource "local_file" "f9" {
 filename = each.value
 for_each = toset(var.filename1)
        content  = "test"
       }
       variable filename1 {
type = list(string)
       default = ["a1","b1","c1"]
}
```

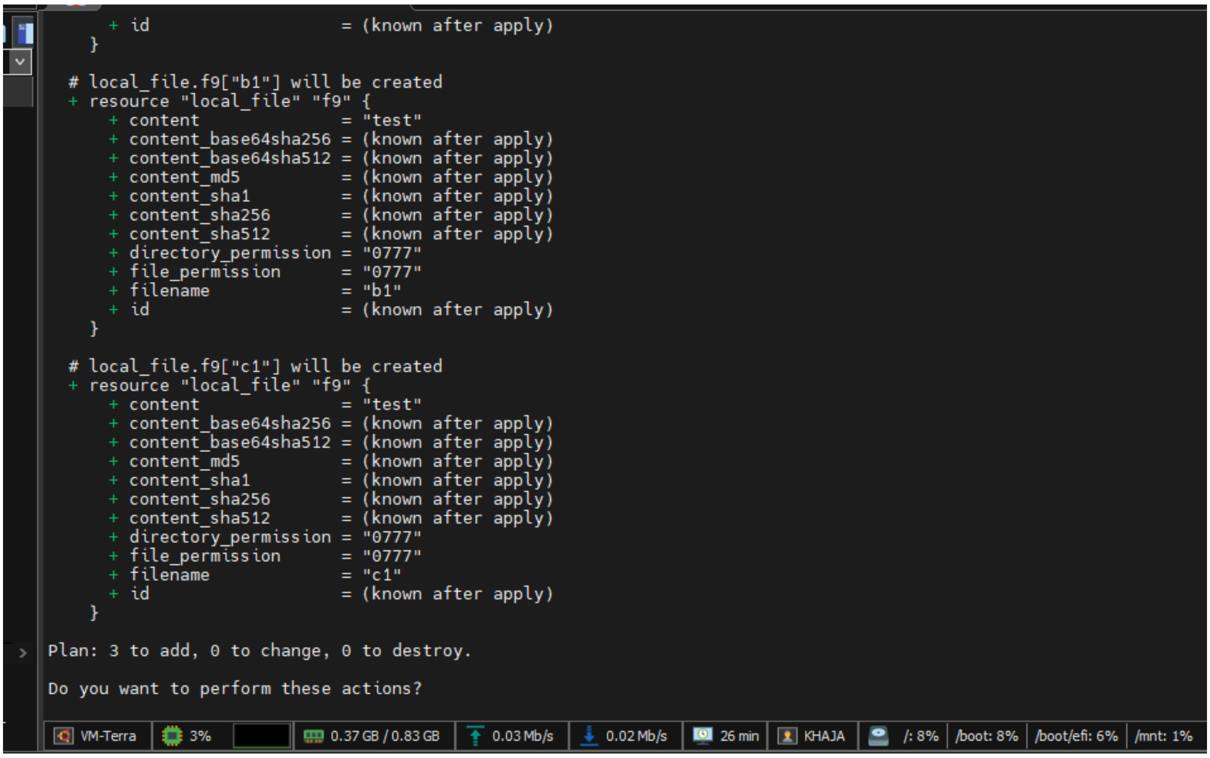


```
                                            }
resource "local_file" "f9" {
                                              filename = each.value
                                              for_each = toset(var.filename1)
                                              content  = "test"
                                            }
                                            variable filename1 {
                                    type = list(string)
                                    default = ["a1","b1","c1"]
                            }
"res.tf" 49L, 998B
```

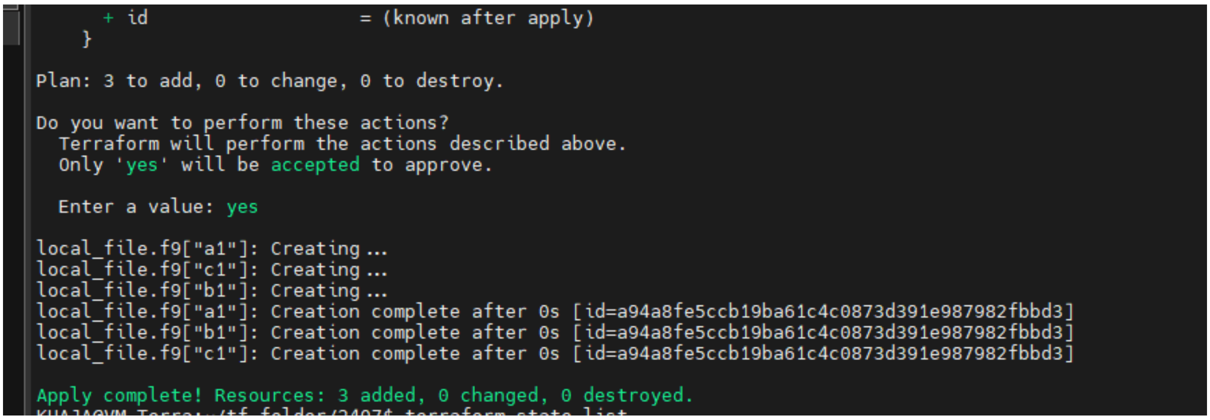
Execute the command terraform apply



```
KHAJA@VM-Terra:~/tf_folder/2407$ terraform apply
local_file.f3: Refreshing state ... [id=7288edd0fc3ffcbe93a0cf06e3568e28521687bc]
local_file.f8[2]: Refreshing state ... [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f8[0]: Refreshing state ... [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f1: Refreshing state ... [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f8[1]: Refreshing state ... [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f2: Refreshing state ... [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]
local_file.f5: Refreshing state ... [id=5b84898d108bab2a9a80e343cc255cc51ce62d3e]
local_file.f4: Refreshing state ... [id=2a3d1a00706cbc6f62772c3aba0fc6b52c4dc328]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # local_file.f9["a1"] will be created
  + resource "local_file" "f9" {
      + content              = "test"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "a1"
      + id                   = (known after apply)
    }

  # local_file.f9["b1"] will be created
  + resource "local_file" "f9" {
      + content              = "test"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
```

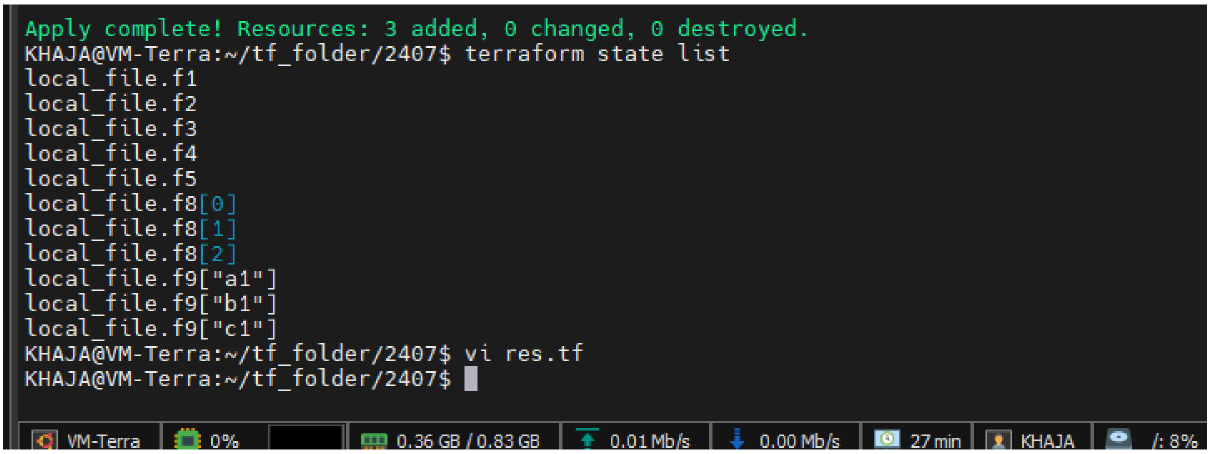


```
      + id                    = (known after apply)
    }

  # local_file.f9["b1"] will be created
  + resource "local_file" "f9" {
      + content               = "test"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5           = (known after apply)
      + content_sha1          = (known after apply)
      + content_sha256        = (known after apply)
      + content_sha512        = (known after apply)
      + directory_permission = "0777"
      + file_permission       = "0777"
      + filename              = "b1"
      + id                    = (known after apply)
    }

  # local_file.f9["c1"] will be created
  + resource "local_file" "f9" {
      + content               = "test"
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5           = (known after apply)
      + content_sha1          = (known after apply)
      + content_sha256        = (known after apply)
      + content_sha512        = (known after apply)
      + directory_permission = "0777"
      + file_permission       = "0777"
      + filename              = "c1"
      + id                    = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
```



```
      + id                    = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.f9["a1"]: Creating...
local_file.f9["c1"]: Creating...
local_file.f9["b1"]: Creating...
local_file.f9["a1"]: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f9["b1"]: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f9["c1"]: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$ terraform state list
```

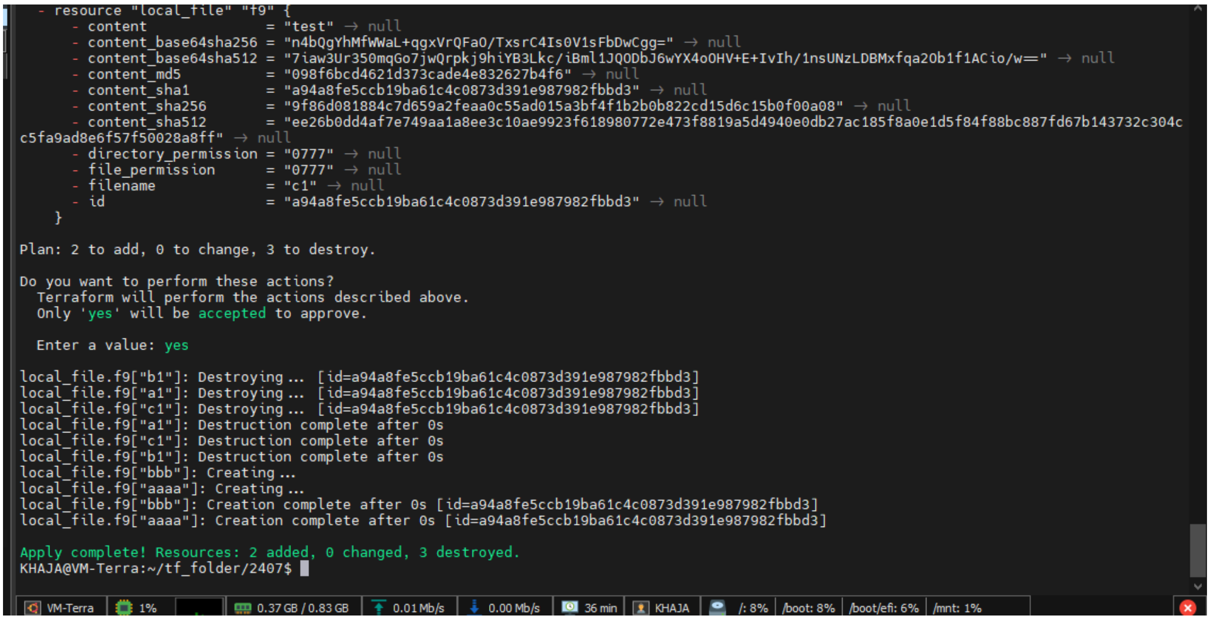
The output will be based on element not the index



```
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$ terraform state list
local_file.f1
local_file.f2
local_file.f3
local_file.f4
local_file.f5
local_file.f8[0]
local_file.f8[1]
local_file.f8[2]
local_file.f9["a1"]
local_file.f9["b1"]
local_file.f9["c1"]
KHAJA@VM-Terra:~/tf_folder/2407$ vi res.tf
KHAJA@VM-Terra:~/tf_folder/2407$ 
```

When we inject the value from outsite like

Terraform apply -var='filename=["aaaa","bbb"]'



```
  - resource "local_file" "f9" {
      - content              = "test" -> null
      - content_base64sha256 = "n4bQgYhMfWWaL+qgxVrQFaO/TxsrC4Is0V1sFbDwCgg=" -> null
      - content_base64sha512 = "7iaw3Ur350mqGo7jwQrpkj9hiYB3Lkc/iBml1JQODbJ6wYX4oOHV+E+IvIh/1nsUNzLDBMxfqa2Ob1f1ACio/w==" -> null
      - content_md5          = "098f6bcd4621d373cade4e832627b4f6" -> null
      - content_sha1         = "a94a8fe5ccb19ba61c4c0873d391e987982fbbd3" -> null
      - content_sha256       = "9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08" -> null
      - content_sha512       = "ee26b0dd4af7e749aa1a8ee3c10ae9923f618980772e473f8819a5d4940e0db27ac185f8a0e1d5f84f88bc887fd67b143732c304c
c5fa9ad8e6f57f50028a8ff" -> null
      - directory_permission = "0777" -> null
      - file_permission      = "0777" -> null
      - filename             = "c1" -> null
      - id                   = "a94a8fe5ccb19ba61c4c0873d391e987982fbbd3" -> null
    }

Plan: 2 to add, 0 to change, 3 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

local_file.f9["b1"]: Destroying... [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f9["a1"]: Destroying... [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f9["c1"]: Destroying... [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f9["a1"]: Destruction complete after 0s
local_file.f9["c1"]: Destruction complete after 0s
local_file.f9["b1"]: Destruction complete after 0s
local_file.f9["bbb"]: Creating...
local_file.f9["aaaa"]: Creating...
local_file.f9["bbb"]: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]
local_file.f9["aaaa"]: Creation complete after 0s [id=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3]

Apply complete! Resources: 2 added, 0 changed, 3 destroyed.
KHAJA@VM-Terra:~/tf_folder/2407$
```

3.for expressions (inside locals or variables)

variable "cities" {

  default = ["delhi", "mumbai", "hyd"]

}

output "upper_cities" {

  value = [for city in var.cities : upper(city)]

}

Result: ["DELHI", "MUMBAI", "HYD"]